PSP/TSP – 6 시그마 도구 적용 방법론에 관한 연구

박영규, 최호진, 백종문
한국정보통신대학교 공학부
e-mail : {youngkyupark, hjchoi, jbaik}@icu.ac.kr

A Study on the Application of Six Sigma Tools to PSP/TSP

Youngkyu Park, Hojin Choi, Jongmoon Baik
Information and Communications University, School of Engineering

요 약

CMM/CMMI 와 같은 프로세스 모델의 등장으로 소프트웨어 프로세스 개선에 대한 원리와 방법은 인식하였으나 현실에 적용하여 성과를 창출하기에는 상당한 어려움을 겪어왔다. 이러한 문제를 해결하고 개발자와 개발팀 차원에서 CMM/CMMI 의 목표와 프렉티스를 구현하기 위해 SEI(Software Engineering Institute)에 의해 PSP/TSP 가 개발되었다. 이렇게 PSP/TSP 가 개인과 팀차원에서 소프트웨어 개발에 사용될 수 있는 구체적인 기법들을 기술하고 있지만 일반적으로 PSP/TSP 에서 수집되는 메트릭에 대한 분석기법은 여전히 부족하다. 따라서 PSP/TSP 수행시 발생할 수 있는 문제를 방지하고 프로세스가 변경되고 유지 관리될 수 있도록 하기 위해서는 6 시그마의 다양한 통계 기법과 의사 결정도구의 사용이 필요하다. PSP/TSP 는 6 시그마가 성공적으로 적용될 수 있는 정량적인 기반을 개인과 프로젝트 차원에서 제공한다. 이에 대해 6 시그마는 PSP/TSP 에서 식별된 문제의 원인을 파악하고 분석하기 위해 필요한 분석도구와 통계적 기법을 제공하고, 문제를 방지하기 위해 프로세스가 변경되고 유지 관리될 수 있는 프로세스 관리 방법론을 제공한다. 따라서 본 논문에서는 PSP 의 각 프로세스에서 활용될 수 있는 6 시그마 도구를 식별하고 활용 가이드라인을 제시함으로써 개인과 팀 차원에서의 프로세스 개선의 수행을 지원하며 팀 차원에서 발생할 수 있는 이슈를 6 시그마의 분석, 정량화 도구를 사용하여 개인 또는 팀의 성과를 향상할 수 있는 방법을 모색해본다.

1. 서론

소프트웨어 개발 프로세스는 설비 중심의 제조공정과는 달리 눈에 보이지 않으며 해당 소프트웨어를 개발하는 인적자원에 크게 의존하는 창의적, 지적 활동의 프로세스이다. 이러한 무형의 개발 프로세스를 구조화하기 위해서 Waterfall, Prototyping, Spiral, Iterative 등의 다양한 소프트웨어 개발 프로세스 모델이 고안되었으며 SEI 의 Watts S. Humphrey 는 소프트웨어 프로세스를 관리상태에 두는 것과 측정하는 것이 지속적인 개선을 달성하는 토대가 된다고 주장하고 조직차원에서의 프로세스를 정의하고 개선하기 위해 process maturity model framework 를 제안하였다[1]. 하지만 이러한 프로세스 모델을 통해 소프트웨어 프로세스 개선에 대한 원리와 방법은 인식하였으나 현실에 적용하여 성과를 창출하기에는 상당한 어려움이 있었다. 이는 개발모형과 방법론을 통해서 수행해야 할 일은 인식하였으나 개인과 팀 차원에서 이를 구체적으로 구현하기 위한 운영적 차원의 절차와 수단이 부족한 것이 주 원인이 되었다. 이러한 문제를 해결하고 개발자와 개발팀 차원에서 CMM/CMMI [2,3]의 목표와 프렉티스를 구현하기 위해서 PSP/TSP 가 개발되었다[4,5].

PSP 는 개발자에게 자신의 개발 산출물을 스스로 관리하고 개선하도록 하는 자기 학습적 방법론을 제공한다. 이를 통해 개발자는 스스로의 작업을 계획하고 추적할 수 있으며, 자신의 작업결과에 대해 프로세스에서 추출한 개발 시간, 결함, 사이즈에 관한 데이터를 분석하고 검토하는 절차를 통해 개선성과를 평가하고 개선활동의 방향을 제시하게 된다.

본 연구는 정보통신부 및 정보통신연구진흥원의
대학 IT 연구센터 육성, 지원사업의 연구결과로 수행되었음

개발자 개인적인 차원의 PSP 를 토대로 하여 CMM 의 프레임워크를 개발팀 차원에 적용함으로써 소프트웨어 품질개선과 생산성 향상을 실현하기 위한 방법론이 TSP 이다. 개발자의 설계에서 테스트까지의 개발단계만을 다루는 PSP 와는 달리 TSP 는 프로젝트 차원에서 수행되는 모든 활동 계획과 측정 작업을 정의하며 CMM 의 KPA 를 실행하기 위한 팀의 수립과 팀 활동을 전개하는 영역으로 확장된다.

비록 PSP/TSP 가 개인과 팀 차원에서 소프트웨어 개발에 사용될 수 있는 구체적인 기법들을 기술하고 있다 하더라도 일반적으로 PSP/TSP 에서 수집되는 메트릭에 대한 분석기법은 여전히 부족하며, 따라서 PSP/TSP 수행시 발생할 수 있는 문제를 방지하고 프로세스가 변경되고 유지 관리될 수 있도록 하기 위해서는 6 시그마의 다양한 통계 기법과 의사 결정도구의 사용이 필요하다.

PSP/TSP 는 6 시그마가 성공적으로 적용될 수 있는 정량적인 기반을 개인과 프로젝트 차원에서 제공한다. 이에 대해 6 시그마는 PSP/TSP 에서 식별된 문제의 원인을 파악하고 분석하기 위해 필요한 분석도구와 통계적 기법을 제공하고, 문제를 방지하기 위해 프로세스가 변경되고 유지 관리될 수 있는 프로세스 관리 방법론을 제공한다. 결론적으로 6 시그마와 PSP/TSP 는 상호 보완적인 관계에 있기 때문에 양자가 동시에 추진될 때 전략적, 관리적 목표 달성에 시너지 효과를 발휘한다고 볼 수 있다. 따라서 본 논문에서는 PSP 개개의 프로세스 영역에서 활용될 수 있는 6 시그마 도구를 식별하고 활용 가이드라인을 제시함으로써 개인과 팀 차원에서의 프로세스 개선의 수행을 지원하며 팀 차원에서 발생할 수 있는 이슈를 6 시그마의 분석, 정량화 도구를 사용하여 개인 또는 팀의 성과를 향상할 수 있는 방법을 모색해본다.

2 장에서는 PSP 와 6 시그마에 대한 기본개념을, 3 장에서는 PSP 와 6 시그마의 통합 방법에 대해서 기술한다. 4 장에서는 결론 및 향후 연구에 대해서 기술한다.

2. 기본 개념

2.1 Personal Software Process

1980 년대 미국방성(Department of Defense)에서는 소프트웨어 품질과 관련된 문제들에 대한 비용손실을 줄이기 위해 Carnegie Mellon University(CMU)의 SEI 에 개선책 연구를 의뢰한 결과 소프트웨어 업체의 개발 프로세스를 개선하기 위한 Capability Mutuality Model(CMM)이 개발되었고 그 후 소프트웨어 개발자 개인 능력을 향상시키고자 SEI 에서 Watts S. Humphrey 가 1990 년 중반 PSP (Personal Software Process)를 제안, 개발자들이 자신에게 맞는 개발방법을 찾고 발전시킬 수 있는 개발 지원 프로세스를 구축하게 되었다. 그리고 PSP 를 조직 내에 정착시킬 수 있도록 1997 년에 TSP(Team Software Process)를 추가로 제안함으로써 팀/개인단위로 경험에 의존한 소프트웨어 개발관행을 정형화하고 조직과 개인간의 유기적인 관계를 바탕으로 지속적으로 개선시킬 수 있는 모델이 성립되었다.

PSP 는 소프트웨어 개발 방법을 훈련 받아 각자의 역량을 향상시키고 체계적이고 잘 정형화된 소프트웨어 개발을 위한 가이드역할을 한다. 또한 PSP 는 개인별로 소프트웨어 개발자의 업무를 예측하고 계획하는 방법과 계획에 따라서 업무 수행 능력을 측정하고 기록하는 방법, 프로그램의 품질을 개선하는 방법을 제시한다. PSP 의 주요목표는 소프트웨어 개발과정에서 개인별로 프로젝트 계획수립을 통해 예상된 시간에 주어진 개발 업무를 최소한의 결함으로 정확하게 완료할 수 있도록 이에 적합한 지침 활동을 제시하는 것이다. PSP 학습 과정을 통해 소프트웨어 개발자는 자신의 습관과 능력을 손쉽게 판단할 수 있고 프로그램의 결함발생률을 줄이면서 양질의 소프트웨어를 개발하는 방법을 배울 수 있다.

2.2 6 시그마

6 시그마는 1987 년 모토로라(Motorola)의 마이클 해리(Mikel J. Harry)에 의해 제창된 통계적 기법을 활용한 품질 개선혁신활동이다. 6 시그마는 원래 통계적인 측정단위로 표준편차를 의미하는데, 프로세스를 정량적으로 평가하고 프로세스 변동을 줄이고자 품질 개선을 위한 경영전략으로 발전하여 구체화되었다[6]. 즉 6 시그마는 일반적으로 경영의 전 부문에서 6 시그마 품질목표를 달성하기 위해 수집된 프로세스의 주요 데이터들을 6 시그마 지원도구들을 사용하여 결함의 원인을 측정, 분석한 뒤 결함을 제거하는 체계적인 품질혁신활동으로 정의할 수 있다[7]. 6 시그마는 프로세스의 낭비를 제거하고 결함과 변동을 줄임으로써 품질 향상이라는 목표를 달성하고자 한다. 각 분야의 선도 기업들은 설계에서부터 제조, 영업, 마케팅, 엔지니어링 등에 걸친 모든 프로세스에 6 시그마 방법론을 적용하고 있다.

3. PSP 와 6 시그마의 통합

PSP 는 소프트웨어 개발자에게 정의된 프로세스를 사용하는 방법, 시간을 추적하고 결함을 기록하는 방법, 프로그램의 크기를 측정하는 방법, 프로젝트의 계획을 수립하고 개발기간과 프로젝트의 크기를 예측하며 달성도(earned-value)를 추적하는 방법 등을 제공하여 스스로 자신의 작업을 관리하고 개선시킬 수 있도록 도와준다. 이에 6 시그마는 PSP 프로세스에서 측정된 데이터를 분석, 평가하여 프로세스 개선을 위한 필요한 효과적인 방법론을 제시한다. 즉 6 시그마는 PSP 활동에서 개인의 개발 프로세스 능력에 영향을 미치는 원인변수를 파악하고 데이터들의 분석을 통해 이들의 연관관계를 도출, 분석하여 필요 시에 수정조치를 취하여 능력을 개선시킬 수 있는 방법론을 제시한다.

다음 장에서는 PSP 프로세스영역과 6 시그마 도구의 매핑 테이블을 통해서 각 단계에서 적용될 수 있

는 6 시그마도구와 도구의 용도를 알아보고 다음으로 실제 적용 프로세스를 예를 들어서 자세히 살펴본다.

3.1 PSP 프로세스영역과 6 시그마 도구의 매핑

다음 표는 PSP 각 프로세스에 사용될 수 있는 6 시그마 도구를 매핑한 테이블이다. 이 테이블은 각 PSP 프로세스의 세부 단계에 적용될 수 있는 6 시그마 도구와 용도를 설명하고 있다. PSP 프로세스가 진행이 될수록 이전에 사용한 프로세스에 새로운 메트릭이 추가되기 때문에 6 시그마 도구 역시 PSP 프로세스가 진행될수록 기존에 사용된 도구에 새로운 6 시그마 도구가 추가된다.

<표 1> PSP 프로세스와 6 시그마 도구 매핑 테이블

Process	Phase	6 시그마 도구	용도
PSP0	Planning	상관분석, 회귀분석	새로운 프로그램의 개발 시간 예측
	Development	파레토분석	전체 결함 중 결함 유형당 차지하는 비율 분석
	Postmortem	상관분석, 회귀분석	새로운 프로그램의 개발 크기와 개발 시간 예측
PSP0.1	Planning	상관분석, 회귀분석	새로운 프로그램의 신규 및 수정 LOC 예측
	Development	다중회귀분석	현재 프로그램의 신규, 재사용, 수정된 LOC 데이터들의 분포에 대한 분석과 새로운 프로그램의 LOC 예측
	Postmortem	PSP0 의 사후(postmortem)단계 내용과 동일	
PSP1	Planning	PROBE	새로운 프로그램의 LOC 와 개발시간의 예측
	Development	PSP0.1 의 개발(development)단계 내용과 동일	
	Postmortem	2-sample t 검정법	신규 및 수정 LOC 에 대한 LOC/Hour 계획값의 정확도에 대한 검토
PSP1.1	Planning	PROBE	새로운 프로그램의 개발시간 예측
	Development	PSP1 의 개발(development)단계 내용과 동일	
	Postmortem	상관분석	CPI(Cost-Performance Index)값의 도출
PSP 2	Planning	막대그래프	각 단계별로 주입되고 제거되는 결함수를 전체 결함수의 누적된 To Date%를 통해 분석
	Development	막대그래프, 파레토분석	형성된 결함수와 제거된 결함수의 실제 데이터를 결함유형별로 우선순위를 매겨 정리한 후 검토를 통해 제거/누락된 결함수를 비교하여 검토 체크리스트 업데이트 수행
	Postmortem	단순회귀분석, 런차트	프로세스 양품률(process yield) 계산
PSP2.1	PSP2 내용과 동일		
PSP3	Planning	PSP2 의 계획(planning)단계 내용과 동일	
	High-level design	특성요인도(설계결함)	이슈 추적 로그에서 프로그램 개발 전반에서 일어나는 문제 체크시 문제 해결책을 위해 특성요인도를 사용하여 보다 효과적으로 문제에 대응할 수 있는 방안 마련
	High-level design review	특성요인도(설계결함)	
	Development	특성요인도(이름/유형 결함)	
	Postmortem	특성요인도(결함제거 율저하)	

3.2 PSP 프로세스에 6 시그마 적용의 실제 예

다음은 PSP1 프로세스의 사후(postmortem)단계에서 2-sample t 검정법을 사용하여 분석하는 예제이다. 프로젝트 계획요약서에 신규 및 수정 LOC 를 총 개발시간으로 나눈 값으로 계획, 실제, 누적 LOC/Hour 데이터를 입력하게 되는데 계획에 대한 LOC/Hour 와 누적된 LOC/Hour 데이터의 차이를 분석함으로써 신규 및 수정 LOC 에 대한 LOC/Hour 계획값의 정확도에 대해 검토할 수 있다.

[예제] 과거부터 프로그램 별 신규 및 수정 LOC 에 대한 계획, 누적된 LOC/Hour 데이터 값들의 차이를 측정, 분석한다.

[단계 1] 예를 들어 계획된 총 신규 및 수정 LOC 가 169LOC 이고 예상 개발시간이 380 분, 누적된 총 신규 및 수정 LOC 가 492LOC 이며 누적된 개발시간은 1663 분이라면 총 신규 및 수정 LOC 에 대한 계획 LOC/Hour 는 다음과 같이 계산할 수 있다.

계획 LOC/Hour = 60 * 169/380 = 26.7
누적 LOC/Hour = 60 * 492/1663 = 17.8

[단계 2] 과거부터 프로그램 별 신규 및 수정 LOC 에 대한 계획, 누적된 LOC/Hour 데이터 값들을 표로 정

리한다.

<표 2> 계획과 누적 LOC/Hour

계획 LOC/Hour(A)					
16.4	20.2	23.4	26.7	30.3	35.2
누적 LOC/Hour(B)					
12.3	14.3	16.2	17.8	20.4	26.7

두 모평균이 동일하다는 가설을 2-sample t 검정법을 사용하여 5%의 유의수준 하에서 검정할 수 있다.

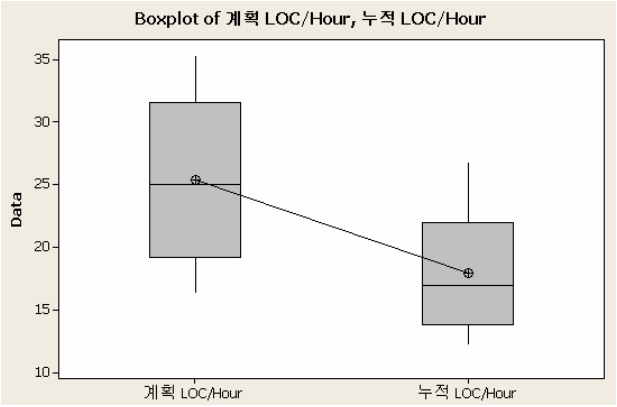
$H_0: \mu_1 = \mu_2$ (귀무가설)

$H_1: \mu_1 \neq \mu_2$ (대립가설)

[단계 3] 데이터를 입력한다.

[단계 4] 검정하고자 하는 변수들을 입력한다.

[단계 5] 표본수가 작을 때 두 그룹에 대한 모평균의 차이를 검정하기 위해 2-sample t 통계기법을 적용하여 측정결과와 Dot plot, Box plot 등의 그래프 결과로 가설을 검정한다.



(그림 1) 계획 LOC/Hour, 누적 LOC/Hour Box plot

<표 3>계획과 누적 LOC/Hour에 대한 2-sample t 검정 결과

	N	Mean	StDev	SE Mean
계획 LOC/Hour	6	25.37	6.84	2.8
누적 LOC/Hour	6	17.95	5.12	2.1

(N: 표본수, Mean: 평균, StDev: 표준편차, SE Mean: 평균의 표준오차)

Difference = μ u (계획 LOC/Hour) - μ u (누적 LOC/Hour)
Estimate for difference(두 항목의 평균값의 차이): 7.41667
95% CI for difference(95% 신뢰구간): (-0.47032, 15.30365)
T-Test of difference =0 (vs not =):
T-Value = 2.13
P-Value = 0.062
DF = 9

[단계 6] P 값이 0.062 로 유의수준 0.05 보다 크므로 양측검정에 의해 귀무가설을 기각할 수 없다. 위 Box plot 결과에서 볼 수 있듯이 '계획 LOC/Hour'과 '누적 LOC/Hour'항목의 자료값들의 평균의 차이가 현저한 차이가 나타나지 않음을 알 수 있다. 따라서 프로그램별 계획 LOC/Hour 와 누적 LOC/Hour 차이가 난다는 명확한 근거를 제시할 수 없다. 즉, 계획한 값이 실제 값과 비교하여 볼 때 정확도가 낮지 않다는 뜻으로 분석할 수 있다.

4. 결론 및 향후 연구

6 시그마는 고객이 체감하는 1 백만분의 3.4 개에 가까운 무결함(Zero Defect) 수준의 제품과 서비스를 설계하고 제공하여 고객만족을 극대화하기 위한 업무 프로세스 개선의 프레임워크를 제공한다. 따라서 소프트웨어 프로세스의 측면에서 데이터 측정을 중시하는 PSP/TSP 에 6 시그마를 적용하는 것은 무결점의 소프트웨어를 개발하고 무결점의 IT 서비스를 창출하기 위한 지속적 프로세스 개선 노력을 가속화하고 체질화하기 위해 기존 6 시그마의 문제 해결 도구와 통계적 기법을 적용하는 것이라 할 수 있다. 이러한 측면에서 본 연구에서는 PSP 관점에서의 6 시그마의 적용영역을 도출하고 예를 들어 설명하였다. 하지만 팀 차원에서 발생할 수 있는 이슈를 6 시그마의 분석, 정량화 도구를 사용하여 개인과 팀의 성과를 향상 시켜야만 프로젝트 및 조직 차원에서 6 시그마 적용을 위한 기반이 마련될 수 있다. 따라서 TSP 관점에서 발생할 수 있는 이슈를 6 시그마의 분석, 정량화 도구를 사용하여 개인 또는 팀의 성과를 향상할 수 있는 방법에 대한 연구를 현재 진행 중이다.

참고문헌

[1] Paulk, M. C., Curtis, B., Chrissis, M. B., et al. Capability Maturity model for software. CMU/SEI-91-TR,24, August 1991.

[2] Mark C. Paulk, B. Curtis, M. B. Chrissis and et al., "Capability Maturity model for Software", Software Engineering institute, CMU/SEI-91-TR-24, DTIC number ADA240603, August 1991.

[3] CMMI, "Capability Maturity Model Integration", Software Engineering Institute, Carnegie Mellon University, 2002.

[4] The Personal Software Process (PSP), Humphrey, Watts S. CMU/SEI-2000-TR-022, ESC-TR-2000-022, December 2000.

[5] The Team Software Process (TSP), Humphrey, Watts S., CMU/SEI-2000-TR-023, ESC-TR-2000-023, December 2000.

[6] Pete Pande, Larry Holpp, *What is Six Sigma?*, McGraw-Hill, 2002.

[7] 이혜영, 최호진, 백종문, 소프트웨어 프로세스의 6 시그마 적용 및 향후 전망, 정보과학회지 특집호 12 월, 2005.